# IDENTIFICATION PAR RADIO FREQUENCE LONGUE DISTANCE RFID

*Richard Kemgo*
richardkemgo@yahoo.fr

*Romual Sitchueng*
sitchueng2000@yahoo.fr

**RÉSUMÉ**

RFID : méthode utilisée pour stocker et récupérer des données à distance en utilisant des balises métalliques, les « Tag RFID ». Ces balises qui peuvent être colées ou incorporées dans des produits sont composées d'une antenne et d'une puce électronique, réagissent aux ondes radio et transmettent des informations à distance. Cette technologie est censée, à long terme, remplacer les codes barres.

## 1. INTRODUCTION

L'abréviation RFID signifie « *Radio Frequency IDentification* », en français, « *Identification par Radio Fréquence* ». Cette technologie permet d'identifier un objet, d'en suivre le cheminement et d'en connaître les caractéristiques à distance grâce à une étiquette émettant des ondes radio, attachée ou incorporée à l'objet. La technologie RFID permet la lecture des étiquettes même sans ligne de vue directe et peut traverser de fines couches de matériaux (peinture, neige, etc.). Le système RFID soumis à notre étude est composé d'un *émetteur* (« *OBID i-scan UHF* ») ou lecteur qui capte et transmet les informations, et d'un *récepteur* ou transpondeur (composée d'une puce reliée à une antenne, encapsulées dans un support « *RFID Tag* »).

L'objectif de notre travail est de paramétrer le lecteur qui nous est donné pour que à terme, qu'on puisse lire et transmettre à un ordinateur les données liées à chaque puce adapter au lecteur.

Nous allons successivement détailler la partie électronique et informatique, ensuite nous allons présenter les différents outils utilisés pour la réalisation de l'étude, la technique ayant permis la réalisation de l'étude, l'obtention des résultats, enfin on conclura tout en dégageant les améliorations possibles.

## 2. PARTIE ELECTRONIQUE

C'est l'une des parties essentielle de notre travaille car elle est responsable de la mise en place des composants nécessaires au bon déroulement du projet et du traitement des signaux. Dans le cadre de notre travaille, nous avons reçus un module complet (lecteur et transpondeur) avec un dispositif électronique déjà fonctionnel. Nous allons dans cette partie, simplement décrire de manière succincte ce qui constitue l'électronique du module RFID que nous avons reçus et présenter ces différentes caractéristiques.

### 2.1. Description et protocoles de communication

Le lecteur de notre module RFID est un produit de l'entreprise *FEIG electronic*. Nous avons choisit le lecteur de type « *OBID i-scan UHF*» qui a une électronique très élaborée et robuste, et s'alimente à 24V. Il possède une entrée/sortie USB, série (RS232), et peu aussi communiquer par le protocole TCP/IP. Nous avons choisit de communiquer par le port série (RS232) et pour cela nous avons du mettre en évidence ce port série à travers un câble de deux mètres constitués de trois fils (Rx, Tx, 5V). L'une des extrémités est connectée au port RS232 du lecteur et l'autre à un DB9 femelle dans le but de le connecter facilement à un ordinateur.

Nous avons aussi mis en évidence un câble réseaux (RJ-45) pour une éventuelle communication à partir du protocole TCP/IP. L'étiquette radiofréquence (*transpondeur*), est composée d'une puce reliée à une antenne, encapsulées dans un support (*RFID Tag*). Elle est lue par le lecteur qui capte et transmet l'information. Les étiquettes que nous utilisons sont de type passives, car utilise l'énergie propagée à courte distance par le signal radio de l'émetteur. Ces étiquettes à moindre coût sont généralement plus petites et possèdent une durée de vie quasi-illimitée. En contrepartie, elles nécessitent une quantité d'énergie non négligeable de la part du lecteur pour

pouvoir fonctionner. Le lecteur « *OBID i-scan UHF*» émet à une fréquence comprise entre 867 et 869 MHZ pour une distance de lecture de 3m.

## 3. PARTIE INFORMATIQUE

C'est la partie qui a constituée l'essentielle de notre travaille car le but ultime de notre projet est de pouvoir identifier les puces électroniques « RFID tag » à partir du lecteur *OBID i-scan UHF*. Pour cela, nous devons implémenter notre lecteur de manière à ce qu'il puisse détecter une puce électronique appropriée et indiquer par exemple l'objet qui lui est associé. Le lecteur *OBID i-scan UHF* à l'avantage de s'adapter à plusieurs langages de programmation parmi lesquels : *Java, C++, C, Labview, VB.*

### 3.1. Language utilisé

Nous avons utilisé le langage C++ car il allie simplicité et puissance et c'est aussi un langage très rependu ce qui est un avantage pour nous d'avoir une documentation fournie et assez riche au cas où nous rencontrerions des difficultés. Nous avons utilisé comme logiciel de programmation *Visual studio 2010* car nous l'avons acquis facilement et il est aussi très rependu.

### 3.2. Particularité du lecteur OBID

Le lecteur *OBID i-scan UHF* a la particularité d'avoir des fonctions déjà toutes faites (par la firme de fabrication) pour chaque connectivité (RS232, USB, Ethernet) dont il faut comprendre le fonctionnement afin de pouvoir en faire ce que l'on souhaite. Nous avons utilisé la connexion RS232 car les fonctions qui lui sont associées sont simples et facile à implémentées. La documentation qui est associée à ce type de connexion est connue sur le nom de *FECOM* dans la documentation générale du lecteur.

### 3.3. Detail de quelques fonctions utilisées

*FECOM* est constitué de nombreuses fonctions permettant par exemple de transmettre ou de recevoir les données, de détecter les ports com. actifs ou non, d'envoyer des messages d'erreur en fonction de l'erreur détectée etc.… en bref ce sont des fonctions qui permettent d'échanger des données entre le lecteur et l'ordinateur. Les fichiers (librairie) nécessaires à l'adaptation de ces différentes fonctions à l'environnement de Visual studio sont fournis par la firme de fabrication. Il s'agit des fichiers FECOM.dll, FECOM.lib, FECOM.h.

Une fois le logiciel de programmation installé avec les librairies nécessaires pour implémenter le lecteur, nous avons procédé à l'analyse des fonctions permettant de communiquer avec le lecteur. Nous avons commencé par mettre en action le programme de démonstration fourni par le fabriquant afin de voir un exemple d'échanges de données entre ordinateur, lecteur et *tag RFID.* L'échange de données ce fait en plusieurs étapes :

- **première étape**

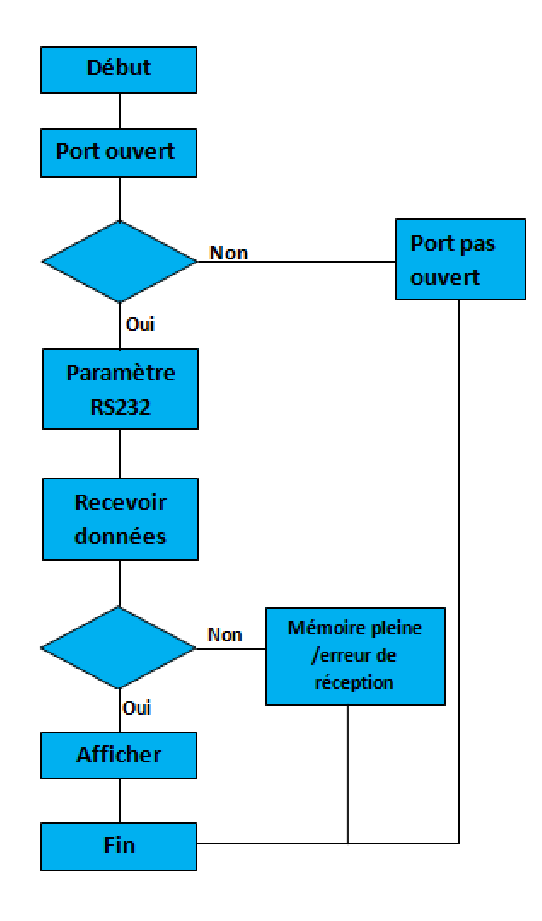
Elle consiste à détecter le ou les ports actifs susceptibles d'être connecté à un lecteur. La fonction qui permet de la réaliser est « *FECOM_openport (cportNr)* » (voir code source) cette fonction prend comme paramètre le numéro du port.

- **Deuxième étape**

Une fois le port auquel est connecté le lecteur est détecté, il doit être paramétré (timeout, vitesse) pour une communication série adéquate avec le lecteur : la fonction qui permet de la réaliser est « *FECOM_setportPara (handle, para, cvalue)* » (voir code source) la fonction prend en paramètre le numéro du port, la vitesse de communication et le timeout. C'est seulement après les étapes précédentes qu'on procède au transfère de données.

- **Troisième étape**

Ce qui nous intéresse dans le transfert de données c'est la lecture des codes fournis par les *tags RFID.* Pour cela, le lecteur doit pouvoir recevoir le code du *tag RFID* afin de l'afficher sur l'ordinateur. La fonction qui permet de recevoir les données du lecteur est *« FECOM_receive (handle, cRecbuf, 256) »* (voir code source). Cette fonction prend en paramètre le numéro du port, une variable pour la réception du code et la taille maximum de la donnée à recevoir. L'organigramme algorithmique ci-dessous nous montre les différentes étapes de communication entre l'ordinateur, le lecteur et le *tag RFID.*




## 4. MATERIEL ET METHODE

Nous allons dans cette partie présenter tour à tour les outils utilisés, les techniques ayant permit de réaliser l'étude, les résultats obtenus et les différentes étapes (dans le temps) de réalisation du projet.

### 4.1. Matériels utilisés

Pour mener à terme notre projet, nous avons fait acheter pour l'essentielle un lecteur de puces RFID de type *OBID i-scan UHF* à la firme *FEIG electronic* spécialisée dans ce domaine. Ce lecteur émet à une fréquence comprise entre 867 et 869 MHZ pour une distance de lecture de 3m. Nous avons aussi utilisé un câble connecter à un DB9 femelle pour la connexion série (RS232) entre le lecteur et un ordinateur (Windows XP). Le logiciel de programmation utilisé est *Visual studio 2010* car il est très rependu et offre une interface d'utilisation simple.

### 4.2. Etude du projet

Pour l'étude du projet, nous avons lu minutieusement la documentation associée au lecteur qui nous a renseigné sur différentes techniques de programmation utilisées pour implémenter le lecteur. Nous avons choisit la communication via le port série car plus aisé à implémenter.

### 4.3. Étapes de réalisation

La réalisation du projet c'est fait durant toute la période allouer au bureau d'étude selon les étapes suivantes :

Du ***04/02/2011 au 04/03/2011*** présentation du projet par les étudiants de première master, choix, commande et réception du lecteur RFID.

Du ***18/03/2011 au 25/03/2011*** mise sous tension du lecteur et communication entre lecteur et ordinateur (test du module), demande à la firme *FEIG electronic* des fichiers (FECOM.DLL, FECOM.H, FECOM.LIB) nécessaires pour l'implémentation du lecteur.

Du ***29/04/2011 au 20/05/2011*** réception des fichiers demandés et création de notre propre application.

## 5. RESULTATS FINAUX

Les résultats obtenus à la fin du bureau d'étude ne sont pas définitifs mais nous permettront d'avancer plus tard dans la réalisation du projet. Nous avons pu mettre en communication le lecteur et l'ordinateur et nous pouvons aussi paramétrer le port série à partir de notre application. Nous avons aussi tenté en vain pour le moment de recevoir le code d'une puce RFID : on reçoit en permanence un message d'erreur (-1030) qui représente une erreur de réception que nous essayerons de résoudre.

L'image suivante représente la console (du logiciel *Visual studio*) qui affiche les paramètres de communication du port série et le message d'erreur qui survient pour le moment lorsqu'on essaye de

recevoir le code d'une puce RFID.

```
F:\visual studio\AppRFID\Debug\AppRFID.exe
ouverture du port avec succès
TimeOut Parameter: 8000
handler Parameter: 1
autre Parameter: Timeout
IRECTPROTLEN: -1030
Return from Recieve:
```

Les images qui suivent représentent une partie du code et un écran qui montre comment on reçoit et change le code d'une puce RFID.

```
// AppRFID.cpp : définit le point d'entrée pour l'application console.
//

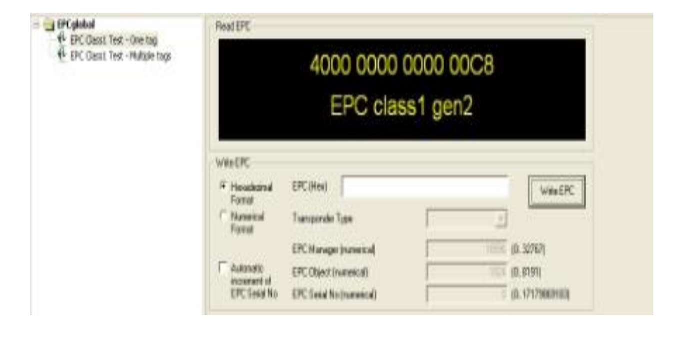
#include "stdafx.h"
#include <windows.h>
//#include "FeCom.h"
#include <fecom.h>
//t#include "FeComDef.h"
#include <windows.h>
#include <tchar.h>
#include <assert.h>
#include <stdio.h>
#include <iostream>
#include <iomanip>

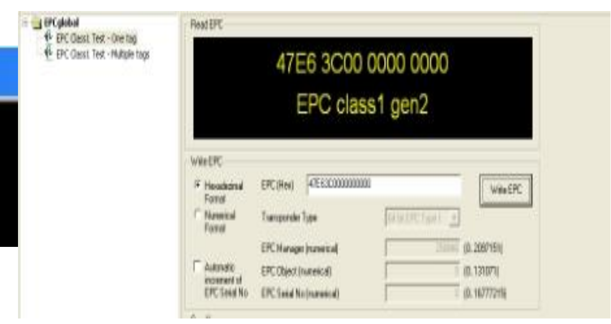
using namespace std;

int main(int argc, _TCHAR* argv[])                    //int main(int argc, char* argv[])
{
    int iRecProtLen;
    char cPortNr[4];
    char cValue[128];
    char* para = "Timeout";
    char* toValue = "8000";
    _itoa( 1, cPortNr, 10 ); // Converti Integer to Char
    UCHAR cRecBuf[256];
    int handle = FECOM_OpenPort( cPortNr ); // COM:1 est ouvert
    if( handle < 0 )
    {


        //cout<<"Error opening the port"<<endl;
        cout<<"erreur d'ouverture du port"<<endl;
    }
    else
    {
        // Communication via COM:1; si possible, les données recues sont localisé dans cRecBuf
```

L'image ci-dessus représente le début du code source de notre application. L'intégralité du code sera jointe en annexe de ce rapport.




Les images ci-dessus représentent deux codes différents (en jaune) pour une même puce. Elle nous présente comment on change le code d'une puce à partir du programme de démonstration. L'intégralité de ces images sera jointent en annexe de ce rapport.

## 6. DIFICULTES RENCONTREES

L'une des principales difficultés rencontrées fut au niveau de l'acquisition de la documentation sur le lecteur RFID car il nous a été livré sans documentation et donc sans manuel d'utilisation. C'est après de nombreuses requêtes auprès de la firme de fabrication qu'un fichier bien sécurisé contenant toutes la documentions du lecteur nous à été envoyer par email. Nous avons aussi rencontré une grande difficulté pour débuter l'implémentation du lecteur car nous n'avions pas toutes les librairies nécessaires et après des recherches sur internet on s'est rendu compte qu'il fallait une fois de plus faire une requête à la firme de fabrication pour obtenir ces librairies et c'est à la dernière semaine du temps alloué au projet que nous l'avons finalement reçu.

## 7. CONCLUSION ET AMELEORATIONS POSSIBLES

Au terme de ce bureau d'étude, nous pouvons dire que nous avons eu une avance qui n'est pas des moindres car nous avons rencontrés de nombreuses difficultés la plus part de temps liées au manque d'informations sur le module. Nous avons pu surmonter une grande partie de ces difficultés et en ce moment ont a déjà pu établir la communication entre le lecteur et l'ordinateur. Mais une difficulté demeure : celle de recevoir effectivement le code fournit par une puce RFID. Pour le moment nous recevons en permanence un message d'erreur liées au dépassement du temps d'attente.

Nous allons prochainement essayer de résoudre ce problème de réception, et améliorer notre code source en créant une interface graphique permettant

d'utiliser plus aisément le lecteur et toutes les ressources possibles de ce lecteur.

Nous pouvons dire en définitive que ce projet nous a permit d'acquérir une expérience supplémentaire dans la gestion d'un projet et de ce rendre compte qu'une analyse bien élaborée et une rigueur sans équivoque nous permet de réaliser un projet sans difficultés majeurs.

## 8. BIBLIOGRAPHIE

[1] http://www.feig.de

[2] http://stackoverflow.com